(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
23 August 2001 (23.08.2001)

PCT

(10) International Publication Number
WO 01/61437 A2

(51) International Patent Classification[7]: G06F 1/00

(21) International Application Number: PCT/US01/04424

(22) International Filing Date: 12 February 2001 (12.02.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/505,890 17 February 2000 (17.02.2000) US

(71) Applicant: GENERAL INSTRUMENT CORPORATION [US/US]; 101 Tournament Drive, Horsham, PA 19044 (US).

(72) Inventors: GOFFIN, Glen; 351 Dublin Pike, Fountainville, PA 18923 (US). BOOTH, Robert; 1700 Rockcress Drive, Jamison, PA 18929 (US).

(74) Agent: KANANEN, Ronald; Rader Fishman & Grauer PLLC, 1233 20th Street, NW, Suite 501, Washington, DC 20036 (US).

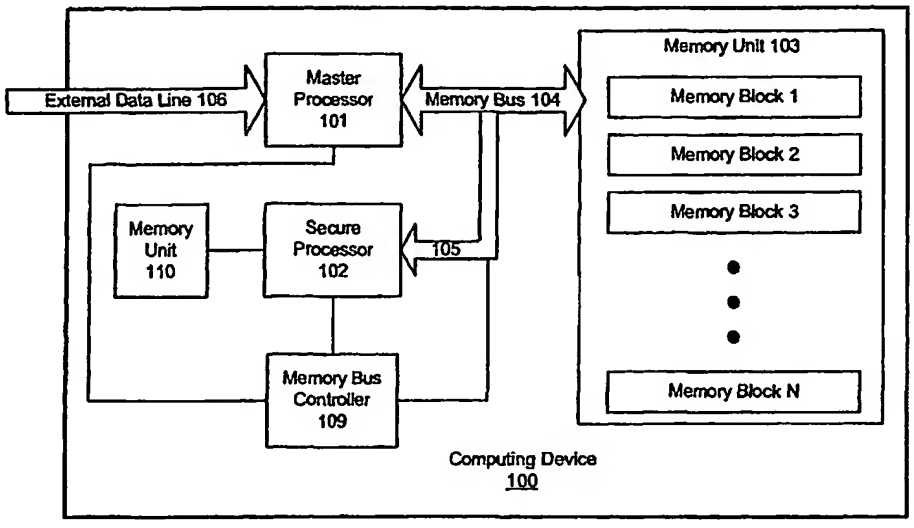
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR PROVIDING SECURE CONTROL OF SOFTWARE OR FIRMWARE CODE DOWNLOADING AND SECURE OPERATION OF A COMPUTING DEVICE RECEIVING DOWNLOADED CODE



(57) Abstract: A method and system for secure downloading of software includes a master processor for receiving downloaded blocks of code which are associated with a computed authentication signature derived from the content of the code block. A secure processor re-computes the authentication signature fromt he content of the downloaded code block and compares the computed signature to that received with the code block. If the signatures do not match or if no signature is appended to the downloaded data, the secure processor determines that the code is from an authorized sender or has been altered during transmission, perhaps to include a virus. The secure processor then takes appropriate protective action.

WO 01/61437 A2

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

TITLE OF THE INVENTION

Method and Apparatus for Providing Secure Control of Software or Firmware Code Downloading and Secure Operation of a Computing Device Receiving Downloaded Code.

FIELD OF THE INVENTION

The present invention relates to the field of transmitting software or firmware programming code to a recipient computing device, such as a set-top terminal. The present invention also relates to the field of controlling the downloading of such code to the computing device in a secure manner such that no virus or alteration in the code is permitted, and such that the code is not freely transferable to other computing devices. The present invention further relates to the field of secure operation of a computing device that executes downloaded program code so as to prevent that computing device from being used to gain unauthorized access to services. The present invention further relates to the field of secure operation of the computing device so as to prevent the execution of a virus or other code that may impair the computing device or cause it to function in an unauthorized manner.

BACKGROUND OF THE INVENTION

Communication in modern society has been greatly enhanced by the advent of such widespread data networks as public telephone systems, the world wide web, and cable and satellite television systems. Particularly with the world wide web and other computer networks, messages and even executable software can be readily transmitted to remote locations. The electronic transmission of

software can be used to commercially distribute such software or to upgrade the existing software of purchasers or data terminals connected to a network.

While such ready access to software and other information is extremely valuable and advantageous, security concerns are also raised. A pervasive problem with computing networks, principally the internet, is the spread of computer viruses that destroy data or impair the operation of computerized systems.

An additional problem is the unauthorized sharing or copying of software. Such copying can be extremely expensive to the owner of the copyright in the software who likely spent significant resources to create the software.

These concerns are also raised in the context of cable television systems. In cable television systems, a set-top terminal or box is typically provided by the cable television company to its subscribers. The set-top terminal is connected, for example, between a cable system outlet and the subscriber's television set or computer. The set-top terminal then receives the cable television signal from the headend facility of the cable system operator.

The set-top terminal may perform any of a variety of functions. For example, the set-top terminal may control the subscriber's access to particular channels on the cable network. If the subscriber has not subscribed to "premium" channels, the set-top terminal may restrict the subscriber's access to those channels. Alternatively, the set-top terminal may enable access to those premium channels to which the customer has subscribed.

Additionally, there is a current trend to combine the various data networks available so that each household or office has a single connection to the information superhighway. An example of this trend is the developing ability of cable television networks to also provide internet access. In such systems, the set-top terminal may be an enhanced terminal with a modem that provides a connection to the internet as well as to the cable television signals. The set-top terminal may operate with the associated television set to provide web browsing capability or may be connected to other computer equipment.

The set-top terminal typically includes software or firmware executed by a central processor of the set-top terminal that enables the terminal to perform the various functions it is called upon to perform. There are at least four principal concerns with regard to the software or firmware in a set-top terminal.

First, it is necessary to prevent a user from altering the software or firmware in the set-top box so as to allow unauthorized access to services available over the cable network for which that user has not paid. The commercial viability of the cable system operator may depend on being able to prevent unauthorized and unpaid access to system services. Such services may include virtually any electronic data service, for example, a range of television channels, premium channels, pay-per-view programming, video-on-demand programming, internet access, electronic mail, telephony, etc.

A related concern is the problem that unauthorized attempts to so modify the software or firmware in a set-top terminal may damage the

terminal or cause it to fail. This may be of particular concern to the cable system operator if ownership of the box is retained by the system operator as is often the case. Additionally, damage may be done to the reputation of the set-top box manufacturer, particularly if the cause of the failure of the set-top box, i.e., illicit modification to the set-top box, is successfully concealed.

The unauthorized modification of the software or firmware of a set-top terminal may also adversely effect the cable system itself. This adverse effect may even extend to causing part or all of the cable system to fail. Such a result may be intentional or unintentional. In either event, the operator of the cable system may be significantly injured by both damage to the cable system and customer dissatisfaction.

Second, the software or firmware in the set-top terminal should be protected from the intentional and malicious introduction of a computer virus that will impair the function of the set-top terminal. This is particularly true where the set-top terminal is being used to connect to the internet.

Third, the software or firmware in the set-top terminal may need to be periodically upgraded or modified as the cable system evolves or as the subscriber requests additional services or the removal of services no longer desired. A preferred means of making such modifications to the set-top terminal is to download the upgraded or modified software or firmware over the cable network itself. This eliminates the need for a technician to visit the location of the set-top terminal. However, it then becomes necessary to provide some security when

the new software of firmware is downloaded so that it is not altered when received and stored in the set-top terminal.

Fourth, the software or firmware in a set-top terminal is generally proprietary in nature and may represent a significant investment. Therefore, the software or firmware in a set-top box should be protected from unauthorized copying, e.g., for use in another set-top terminal.

Therefore, there is a need in the art for a method and system of securing the downloading of software or firmware, particularly to a set-top terminal in a cable television system, so as to ensure that the software or firmware is transmitted as authorized without alteration or the introduction of a virus. There is a further need in the art for a method and system of preventing modification of the software or firmware stored in and executed by the terminal so as to gain unauthorized access to services. There is a further need in the art for a method and system of preventing illicit copying of software or firmware, particularly from a set-top terminal, for some unauthorized use.

SUMMARY OF THE INVENTION

It is an object of the present invention to meet the above-described needs and others. Specifically, it is an object of the present invention to provide a method and system for securing the downloading of software or firmware, particularly to a set-top terminal in a cable television system, so as to prevent unauthorized alterations of, or the introduction of a computer virus to, the software or firmware during transmission. It is a further object of the present

invention to provide a method and system for preventing the unauthorized duplication of software or firmware, particularly in a set-top terminal of a cable television system.

Additional objects, advantages and novel features of the invention will be set forth in the description which follows or may be learned by those skilled in the art through reading these materials or practicing the invention. The objects and advantages of the invention may be achieved through the means recited in the attached claims.

To achieve these stated and other objects, the present invention may be described as a system for secure transmission of programming code to a computing device. In a principal embodiment, the system of the present invention includes: a master processor connected to an input data line for receiving blocks of programming code, each of which is associated with an authentication signature derived from the content of the block of code; a memory unit for storing the blocks of programming code; and a secure processor for independently computing an authentication signature for each block of programming code and matching the computed authentication signature against the authentication signature transmitted with that block of code. If the computed authentication signature fails to match the transmitted authentication signature, the secure processor takes appropriate action, such as purging the unauthenticated block of code from memory or disabling the computing device.

A second, isolated memory unit connected to the secure processor may be provided for storing data needed by the secure processor to compute authentication signatures. The secure processor may

continually re-authenticate the blocks of programming code stored in the memory unit to ensure their continued legitimacy.

The master processor, memory unit and secure processor are all interconnected by a memory bus. To avoid conflicts between the master and secure processors, the secure processor preferably accesses the memory unit only during bus cycles in which the master processor is not using the memory bus. A memory bus controller is provided to police this arrangement and control access to the memory bus by the main and secure processors.

Preferably, the memory unit comprises an interim memory and a static memory. The master processor writes the downloaded blocks of code to the interim memory initially. After the secure processor has authenticated the blocks of code, the blocks of code are transferred to the static memory.

In a preferred embodiment, the computing device is a set-top terminal incorporating the master processor, memory unit and secure processor. The input data line is a connection to a cable television system.

In such an embodiment, an entitlement management message may be associated with the blocks of programming code. The secure processor will then prevent further storage, use or execution of the blocks of programming code unless the entitlement management message matches a predetermined entitlement management message stored by the secure processor. This allows selective set-top terminals to be reprogrammed by a broadcast of new programming code to all the set-top terminals in the system indiscriminately.

As an additional security measure, the blocks of code may be encrypted when received by the master processor. The secure processor would preferably then include a decrypter, for example executable decryption software, for decrypting the encrypted blocks of code.

The present invention also encompasses a method for making secure transmission of programming code to a computing device and, more importantly, for monitoring and insuring the continued integrity of programming code already stored in the computing device. Using the principles inherent in the system described above, the method includes the principal steps of computing an authentication signature for each of one or more of blocks of programming code received or stored in a memory unit with an algorithm, the authentication signature being computed based on the numeric content of that block of programming code; and comparing the computed authentication signature to a transmitted authentication signature transmitted or stored in association with that block of programming, the transmitted authentication signature having been generated with the same algorithm.

The present invention also encompasses a second system for secure transmission of programming code to a computing device. This second system includes: a master processor connected to an input data line for receiving blocks of programming code, each of which is associated with an authentication signature derived from the content of the block of code; a memory unit for storing the blocks of programming code; and a background software task running on the master processor for independently computing an authentication signature for each block of

programming code using the same method as the legitimate sender and matching the computed authentication signature against the authentication signature transmitted with that block of code. As before, if the computed authentication signature fails to match the transmitted authentication signature, the background software task may disable the computing device.

If the memory unit comprises an interim memory and a static memory, the blocks of code may be transferred to the static memory when the background software task has authenticated those blocks of code. If an entitlement management message is associated with the block of programming code, the background software task may prevent further use, storage or execution of the blocks of programming code unless the entitlement management message matches a predetermined entitlement management message stored in the computing device.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings illustrate the present invention and are a part of the specification. Together with the following description, the drawings demonstrate and explain the principles of the present invention.

Fig. 1 is a first embodiment of a computing device, such as a set-top terminal of a cable television system, according to the present invention.

Fig. 2 is a second embodiment of a computing device according to the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Using the drawings, the preferred embodiments of the present invention will now be explained.

Fig. 1 illustrates a computing device (100) according to the present invention. The computing device (100) may be any device which downloads software or firmware from a remote source over a data line (106) and stores the same for later execution. In a preferred embodiment, the computing device (100) is a set-top terminal in a cable television system, and the data line (106) is an external connection to the cable signal provider. Alternatively, however, the data line (106) could be connected to any external port of the computer device or could be an internal data line of the computing device (100) over which someone may introduce programming code to the device (100), particularly to the master processor (101). As used herein, programming "code" includes, but is not limited to, bootcode, BIOS (basic input/output sofware) code, operating system software or firmware, device driver software or firmware, network protocol software or firmware, application software (e.g., web browsers, electronic program guides, e-mail, chat, etc.), web pages and graphics images.

Under the principles of the present invention, the computing device (100) includes two distinct processors. The first is a master processor (101) which may be, for example, a central processing unit. The master processor (101) is the processor which executes object code to perform the functions of the computing device (100). The second processor is a secure processor (102) that is used to authenticate code downloaded over the data line (106) so as to prevent the introduction of a virus

or code which has been altered during transmission, or an unauthorized code object. The secure processor (102) also may continually re-authenticate code previously received and stored in a memory unit (103) to prevent unauthorized tampering with or additions to the executable code and data resident in the set-top terminal (100).

Under the principles of the present invention, when software or firmware is to be downloaded to the computing device (100), the legitimate sender will break the code being sent into blocks or packets. The content of each block or packet of code data is then used to compute an authentication signature using an algorithm which may require one or more keys as will be understood by those skilled in the art.

The data blocks are then transmitted, in association with the computed authentication signature, to the master processor (101) over the external data line (106). The master processor (101) using a memory bus (104) writes the downloaded code to a memory unit (103). The memory unit (103) may be any form on non-volatile, long-term electronic data storage device including, for example, an electronic memory device, a magnetic hard drive or an optical or magento-optical disc drive.

The memory unit (103) is divided into memory blocks (Block 1 to Block N). These memory blocks may correspond to physical sections of the memory unit (103) in which data can be stored. The memory blocks may also preferably correspond to the blocks or packets of data into which the software or firmware is partitioned during the downloading.

The secure processor (102) has a connection (105) to the memory bus (104) and thence to the memory unit (103). After the software or firmware has been downloaded into the memory unit (103), the secure processor (102) will then access each memory block. The secure processor (102) is programmed to duplicate the algorithm used by the legitimate code sender to generate the authentication signature of each block of code.

Consequently, the secure processor (102) can authenticate the code received by the master processor (101) and stored in the memory unit (103). As an alternative, the code blocks may be initially received by the secure processor (102) rather than the master processor (101). In such a scenario, the secure processor (102) authenticates the code blocks before storing them in the memory (103). An separate interim memory unit or a specified section of the memory unit (103) may be used to store code blocks being authenticated by the secure processor (102) before those code blocks are cleared for storage in the long-term memory of the memory unit (103).

The authentication process will be now described in more detail. The secure processor (102), using the code in each memory block as an input, re-computes the authentication signature based on the numeric content of the code block and any keys used by the legitimate sender. The authentication signature computed by the secure processor (102) is then compared by the secure processor (102) to the authentication signature transmitted in association with that block of code.

If the signatures match, the code in that memory block is authenticated as having been transmitted by an authorized sender without alteration and without the introduction of a dangerous virus. If, on the other hand, the authentication signatures do not match, or the data block does not include an authentication signature at all, the secure processor (102) determines that the code has been altered, perhaps to include a virus, during transmission or has been sent by an unauthorized sender who has no authority to reprogram the computing device (100). The secure processor (102) can then erase or disable the adulterated memory block or blocks.

Alternatively, the secure processor (102) can disable the entire computing device (100) requiring the intervention of a technician or a signal from the system operator to re-initialize the device (100). This may assist the service provider to identify individuals who have attempted unauthorized alterations to the software or firmware in the computing device (100). Such a feature is particularly of interest if the computing device (100) is a set-top terminal of a cable television system as in the preferred embodiment.

The secure processor (102) may be hard-wired or firm-wired to compute the authentication signatures of the data blocks. For more flexibility, the secure processor (102) may operate by executing instructions in an internal memory device or an external memory unit (110).

Because the response to unauthorized or adulterated programming originates from the relatively isolated secure processor (102) rather than the main processor (101), it is inherently more

difficult to anticipate and defeat. Thus, the system of the present invention which comprises two distinct processors is inherently more secure than previous systems.

In addition to authenticating blocks of code newly received in the computing device (100), the secure processor (102) will also preferably make a periodic sweep of the code blocks in the memory unit (103) to re-authenticate those code blocks thereby insuring that the code in the memory unit (103) has not be altered or modified by an unauthorized party. It is as important to protect the continued integrity of executable programming code already stored in the memory unit (103) as to guard against the introduction of new, unauthorized or adulterated code.

The operation of the secure processor can be kept from interfering with or slowing the operation of the computing device (100) by requiring the secure processor (102) to access the memory blocks of the memory unit (103) during "stolen" bus cycles. This means that the secure processor (102) is only allowed to access the memory unit (103) during bus cycles for which the master processor (101) is occupied with internal processing and is not accessing the memory bus (104).

For this purpose, a memory bus controller (109) is preferably provided. The memory bus controller (109) regulates the access to the bus of the two processors (101) and (102). The memory bus controller (109) may also monitor activity on the bus (104) directly. Preferably, the memory bus controller (109) responds to the master processor (101) so that when the master processor (101) is not using the memory bus (104), the memory bus

controller (109) grants access to the memory bus (104) to the secure processor (102).

With the secure processor (102) stealing bus cycles, the secure processor (102) can be used to constantly authenticate and re-authenticate the code blocks in the memory unit (103) to prevent tampering. No loss of speed or inference with the operation of the main processor (101) results.

As an added security measure, the secure processor (102) can also be programmed to monitor the number of bus cycles it receives in any specified period of time. If the number of bus cycles made available to the secure processor (102) falls below a predetermined threshold, the secure processor (102) can be programmed to assume that the set-top box has been tampered with. The secure processor (102) can then initiate an appropriate response such as shutting down the operation of the set-top box and/or signaling the system operator to advise of the problem and request a service call by an authorized technician.

The present invention can also be used to prevent the unauthorized copying of code in the computing device (100) for use in another computing device. If the second computing device to which the code is being copied includes a secure processor (102), that processor (102) will be unable to authenticate the illicitly copied code without the algorithm and/or keys held by the original secure processor.

Alternatively, if the second computing device to which the code is copied does not authenticate code blocks, the authentication signature embedded in the code can be used, as will be clear to those skilled in the art, to disrupt normal execution of

the code by a computer not programmed to recognize, extract and make use of the authentication signature.

For additional security, the code being downloaded over the data line (106) may be encrypted. The secure processor (102) could then also be used to decrypt the code, block by block, in the memory unit (103). Such encryption will further prevent the unauthorized copying of the code for use elsewhere.

In the application of the present invention to a set-top terminal in a cable television system, the problem may arise in which the set-top terminals of some subscribers who are receiving premium services need to receive upgraded software, while other subscribers receiving other or lesser services need no programming modification. The present invention can be used to address this problem by appending an entitlement management message to the code blocks being transmitted. The entitlement management message may be appended to or embedded in the authentication signature or may be an entirely separate string.

The entitlement management message will indicate which class of set-top terminals are to accept and implement the downloading code. The secure processor (102) can then compare the entitlement management message in the downloading code with a complementary message specific to the terminal (100) in which the secure processor resides (102). The complementary entitlement management message may be stored in memory unit (110).

If the complementary entitlement management message of the secure processor (102) matches that associated with the downloading code, then the

secure processor (102) will proceed to authenticate and allow the further storage, use or execution of the downloaded code. In this way, code can be broadcast over the cable network (106) to all the set-top terminals in the system with only a selected set of set-top terminals accepting and implementing the new code.

The secure processor (102) can also play a role in securing the integrity of the programming code stored in the memory unit (103) as that code is executed. For example, before the master processor (101) can execute stored programming code, the code is transferred from long-term memory, e.g. FLASH memory, to RAM (random access memory). The secure processor (102) can be used to transfer the code from long-term memory to RAM and can re-authenticate each block of code incident to that transfer.

Fig. 2 illustrates a second embodiment of the present invention. The principle behind this embodiment is that a second memory unit for storing downloaded code is provided. With two memory units storing code, one is used as a temporary or interim storage area where downloaded data is written and stored initially until it is authenticated by the secure processor (102). After the secure processor (102) has authenticated the downloaded code, the code is transferred from the temporary storage area to long-term, static storage from whence it is accessed and executed by the master processor (101) as needed.

As shown in Fig. 2, a secondary memory unit (200) is connected (201) to the memory bus (104). The secondary memory unit (200) may be a hard drive or some other memory device and may even be external to the computing device (100). Either memory unit

(200) or (103) may be used as the temporary storage, and either may be used as the long-term storage depending on the particular application.

An alternative approach to the present invention without employing two separate processors is to have a secure software or firmware task running on the main processor (101) as a "background" task. Such a secure task could perform all the functions described above for the secure processor (102), including (a) continual sweeping of the protected memory to validate authentication signatures, (b) disabling the computing device functions after a failure to properly authenticate a memory block, (c) checking for proper entitlements as part of background authentication process thus enabling access control of applications, (d) authentication and entitlement validation at the time downloaded code is received, and (e) possible address resolution and linking of the downloaded code.

If the secure task can be carefully design to resist being hacked into or cracked, then the use of a background secure task provides a simple way to obtain many of the advantages of the present invention without requiring the replacement of existing computing devices to provide devices having the two processor system (101 and 102) described above.

The background task can also be backed up by a "watchdog" task that would timeout if not regularly prompted by the background task. In this way, if the background task is somehow defeated and deactivated, the watchdog task will no longer be prompted and can then take appropriate action such as deactivating the set-top terminal and/or

signaling the system operator via the headend facility. Additionally, the prompting between the background and watchdog tasks can use a secure handshake that would be very difficult to counterfeit.

The preceding description has been presented only to illustrate and describe the invention. It is not intended to be exhaustive or to limit the invention to any precise form disclosed. Many modifications and variations are possible in light of the above teaching.

The preferred embodiment was chosen and described in order to best explain the principles of the invention and its practical application. The preceding description is intended to enable others skilled in the art to best utilize the invention in various embodiments and with various modifications as are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the following claims.

WHAT IS CLAIMED IS:

1. A system for secure transmission of programming code to a computing device, the system comprising:

a data line for receiving blocks of programming code;

a master processor for executing blocks of programming code, each of which is associated with an authentication signature derived from a content of said block of code;

a memory unit for storing said blocks of programming code for use by said mater processor; and

a secure processor for independently computing an authentication signature for each said block of programming code and matching said computed authentication signature against said authentication signature transmitted or stored in association with that block of code.

2. The system of claim 1, wherein if said computed authentication signature fails to match said transmitted authentication signature, said secure processor disables said computing device.

3. The system of claim 1, wherein if said computed authentication signature fails to match said transmitted authentication signature, said secure processor resets said computing device.

4. The system of claim 1, wherein if said computed authentication signature fails to match said transmitted authentication signature, said

secure processor signals for an authorized service call for said computing device.

5. The system of claim 1, wherein said master processor, memory unit and secure processor are interconnected by a memory bus, and said secure processor accesses said memory unit only during bus cycles in which said master processor is not using said memory bus.

6. The system of claim 5, further comprising a memory bus controller for controlling access to said memory bus by said main and secure processors.

7. The system of claim 1, wherein:

said memory unit comprises an interim memory and a static memory;

new blocks of code received over said data line are written to said interim memory unit; and

after said secure processor has authenticated said blocks of code, the blocks of code are transferred to said static memory.

8. The system of claim 1, further comprising an entitlement management message associated with said blocks of programming code, wherein said secure processor will prevent further storage, use or execution of said blocks of programming code unless said entitlement management message matches a predetermined entitlement management message stored by said secure processor.

9. The system of claim 1, wherein:

said computing device is a set-top terminal incorporating said master processor, memory unit and secure processor; and

said input data line is a connection to a cable television system.

10. The system of claim 1, further comprising a second, isolated memory unit connected to said secure processor.

11. The system of claim 1, wherein said secure processor periodically re-authenticates said blocks of programming code stored in said memory unit.

12. The system of claim 1, wherein said blocks of code are encrypted when received by said master processor, said secure processor further comprising a decrypter for decrypting said encrypted blocks of code.

13. A method for making secure transmission of programming code to a computing device, the method comprising:

computing an authentication signature for each of one or more of blocks of programming code newly received or stored in a memory unit with an algorithm, said authentication signature being computed based on a numeric content of that block of programming code; and

comparing said computed authentication signature to an authentication signature transmitted or stored in association with that block of programming code, said transmitted authentication signature having been generated with said algorithm.

14. The method of claim 13, wherein if said computed authentication signature fails to match said transmitted authentication signature, said method comprises disabling said computing device.

15. The method of claim 13, wherein said computing and comparing are performed with a secure processor which is separate from a master processor for executing said blocks of code.

16. The method of claim 15, further comprising accessing said blocks of code in said memory unit with a memory bus for authentication by said secure processor only during bus cycles in which said master processor is not using said memory bus.

17. The method of claim 13, further comprising transferring said blocks of code from an interim memory of said memory unit to a static memory of said memory unit after said computed authentication signature has matched said transmitted authentication signature.

18. The method of claim 13, further comprising:

receiving an entitlement management message associated with said blocks of programming code;

storing and executing said blocks of programming code only if said entitlement management message matches a predetermined entitlement management message stored in said computing device.

19. The method of claim 13, further comprising repeatedly performing said computing and comparing

to re-authenticate said blocks of programming stored in said memory unit.

20. The method of claim 13, further comprising decrypting said blocks of programming code which are received in an encrypted form.

21. A system for secure transmission of programming code to a computing means, the system comprising:

a master processing means connected to an input data line for receiving blocks of programming code, each of which is associated with an authentication signature derived from a content of that block of code;

a memory means for storing said blocks of programming code; and

a secure processing means for independently computing an authentication signature for each said block of programming code and matching said computed authentication signature against said authentication signature transmitted with that block of code.

22. A system for secure transmission of programming code to a computing device, the system comprising:

a master processor connected to an input data line for receiving blocks of programming code, each of which is associated with an authentication signature derived from a content of said block of code;

a memory unit for storing said blocks of programming code; and

a background software task running on said master processor for independently computing an

authentication signature for each said block of programming code and matching said computed authentication signature against said authentication signature transmitted with that block of code.

23. The system of claim 22, wherein if said computed authentication signature fails to match said transmitted authentication signature, said background software task disables said computing device.

24. The system of claim 22, wherein if said computed authentication signature fails to match said transmitted authentication signature, said background software task resets said computing device.

25. The system of claim 22, wherein if said computed authentication signature fails to match said transmitted authentication signature, said background software task signals a service call for said computing device.

26. The system of claim 22, wherein:

said memory unit comprises an interim memory and a static memory;

said master processor writes said downloaded blocks of code to said interim memory unit; and

after said background software task has authenticated said blocks of code, the blocks of code are transferred to said static memory.

27. The system of claim 22, further comprising an entitlement management message associated with each said block of programming code, wherein said

background software task will prevent further storage, use or execution of said blocks of programming code unless said entitlement management message matches a predetermined entitlement management message stored in said computing device.

28. The system of claim 22, wherein:

said computing device is a set-top terminal incorporating said master processor and said memory unit; and

said input data line is a connection to a cable television system.

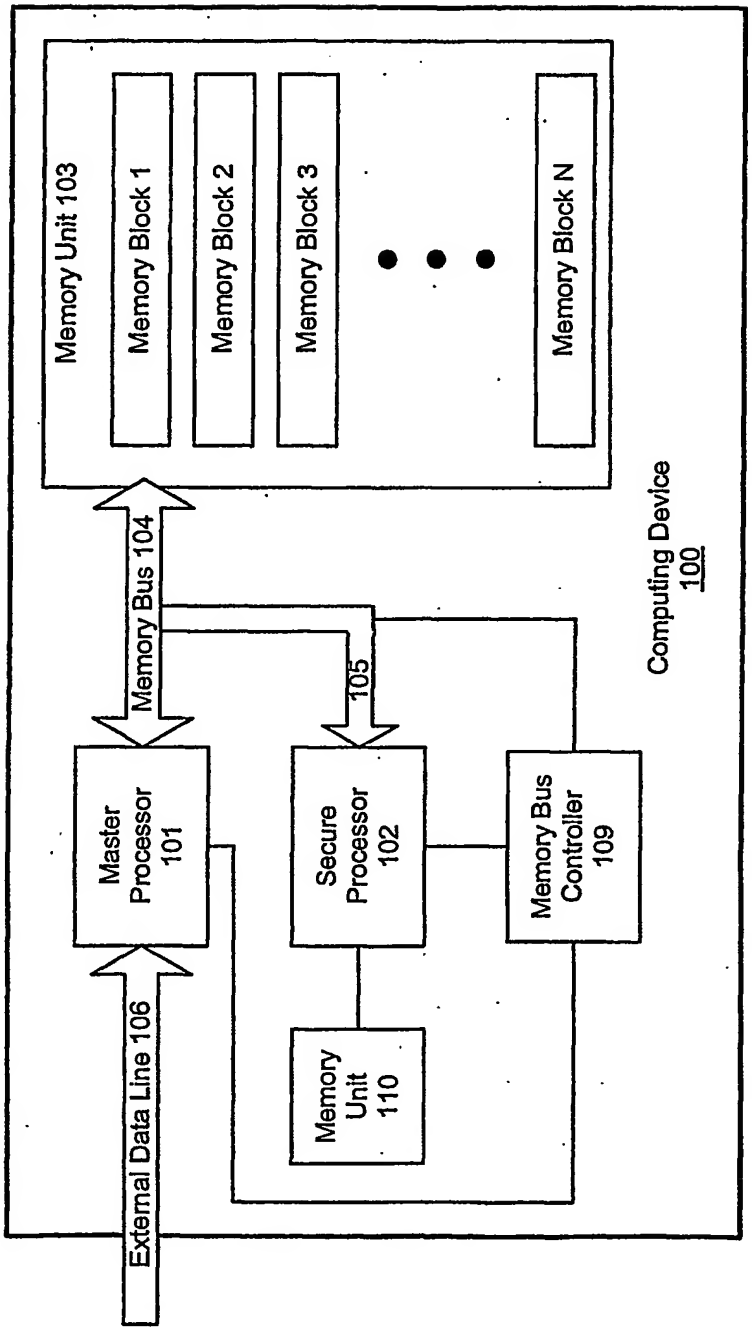

Memory Unit 103
Memory Block 1
Memory Block 2
Memory Block 3
Memory Block N
Computing Device
100
Memory Bus 104
105
Master Processor 101
Secure Processor 102
Memory Bus Controller 109
Memory Unit 110
External Data Line 106

Fig. 1

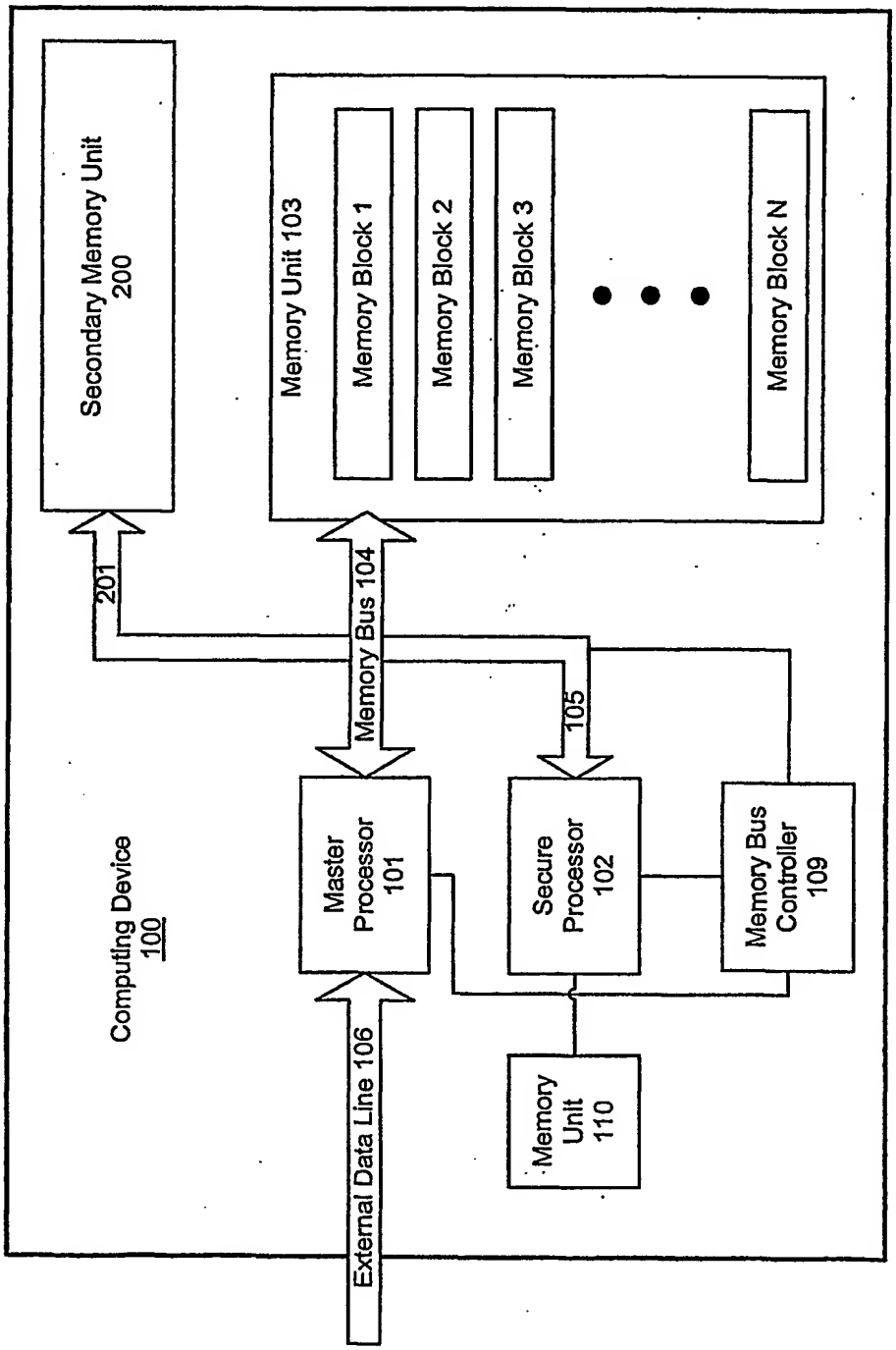

Secondary Memory Unit
200
Memory Unit 103
Memory Block 1
Memory Block 2
Memory Block 3
Memory Block N
201
Memory Bus 104
105
Computing Device
100
Master Processor
101
Secure Processor
102
Memory Bus Controller
109
Memory Unit
110
External Data Line 106

Fig. 2